Using UML to Create a Color Management System

Yui-Liang Chen *, Mei-Chun Lo**, and Chih-Chieh Yu ***

Keywords: Color Management System (CMS), Unified Modeling Language (UML), International Color Consortium (ICC), Object-Orientation, Software Engineering

Abstract: In the past decade, Unified Modeling Language (UML) has become one of the most powerful modeling tools to analyze and design computer systems in the computer specialists' world. However, the experts interested in Color Management System (CMS) lack of such consensus about modeling tools, therefore much redundant work was repetitively done by redesigning and recoding CMS again and again. Until now, there is no complete system analysis and system design of CMS using any modeling tools. Nevertheless, UML is very useful for end-users and computer specialists to communicate with each other by some visual ways. It is also very powerful to rebuild and update the associated system by some forward engineering methods. The goal of this research was to analyze and design an ICC-based CMS using UML. The following works had been accomplished:

- To describe the behavior of CMS as seen by its end users, analysts, and tester in Use Case View.
- To provide the vocabulary of CMS and its solution to end users in Design View.
- To address the performance, scalability, and throughput of the CMS in Process View.
- To manage the configuration of CMS releases made up of independent components and files to produce a running CMS in Implementation View.
- To form the CMS's hardware topology on which the system executes in Deployment View.

Finally, we built a demonstrated CMS and verify its workability by using Java codes.

---

* Dept. of Information Management, E-mail: ychen@cc.shu.edu.tw
** Dept. of Graphic Communications and Technology,
E-mail: mcl@cc.shu.edu.tw
*** Dept. of Graphic Communications and Technology,
E-mail: a851114@gc.shu.edu.tw

1 Introduction

Since the growing of computer technologies, the media of image communications are become complicated. From traditional printing devices to advanced digital devices, the most important color information is not preserved due to the fact of device dependent characteristics of color information during the image communications. Therefore, the restriction of image communications causes people do not rely on the digital image what they see.

Actually, there are some “setting/adjustment” mechanisms of most digital devices. The default settings of most devices are good enough for most common users. However, in case of the need of adjustment, rare of users understand how to tune the settings of devices. Although color adjustment is a very simple thing to human being, the lack of color standards makes it impossible to reproduce the right colors on any devices. In the other word, it is important to have some standards of color adjustment among the variety of digital devices. Moreover, the color standards of different digital devices are the requirements in the recently years.

By the same token, we also need to use a standard to model a framework. The Unified Modeling Language (UML) is objected-oriented and programming language independent modeling tools to describe computer systems in the world. A Color Management System (CMS) is a computer system, which is dedicated for digital device color adjustment. Many existing CMSs were built by different experts from different institutes or companies. However, none of them can be shared each other’s functionality. Every CMS should be rebuilt from the beginning. The objective of our research is to use the modeling tool, UML, to build a framework of CMS for representing color information.

2 Color Management System

In early electronic publish systems, color management was not the major factor. It is because that all input devices (such as scanners), display devices (such as CRT monitors) and output devices (such as digital proofer) were produced by the same manufacturer. In such a close environment, as long as the software and hardware of all devices have been calibrated, it is not difficult to obtain the same colors between the origin and reproduction.

In the recently years, because of the popularity of desktop publishing (DTP) and computer-aided design (CAD), plus the rapid upgrading all kinds of digital devices. Color reproduction is much different from the traditional close environment. In order to transfer and store the color information inter-media without loosing information from the original image, color management module (CMM) as shown in Figure 1 is becoming a solution to solve the open architecture problem. Therefore, a typical color management system is just a combination of CMM components responsible for the communications, connection, and compensation between input devices and output devices. According to its application, it is possible to enhance the capability of color reproduction across media, and to optimize the quality of color consistence. In order to fulfill the color management, the following procedures must be accomplished.

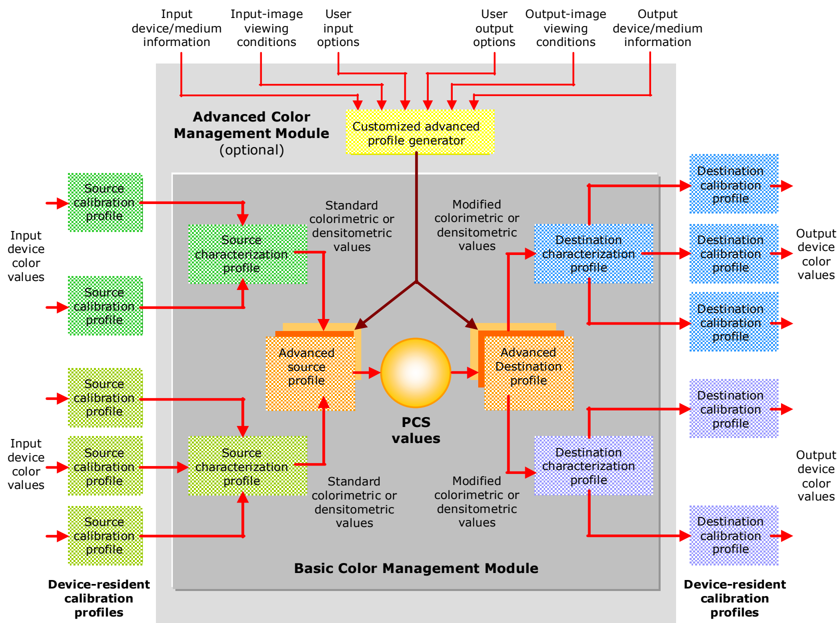


Figure 1. CMM Architecture (modified from Giorgianni and Madden, 1998)

2.1 Device Calibration

Calibration is one of CMS options. Users may measure and record the default profile and different settings of different reproductions through the calibration functions of CMS to compensate colors during executing color transformation. For instance, the white point or other calibrations of monitors are identical to set the parameters of devices onto the default profile. In general, it resides inside device and depends on device itself. Most of experts don’t count this as part of CMS.

2.2 Device Characterization

Characterization is a way to transform color values between different devices. Several typical characterizations are done as follows:

1. The color characterizations of monitors
   Monitors are the intermediate productions between input and output. We have to record the characterization of monitors, or choose the build-in monitors values in the CMS.
2. The characterizations of scanners or other input devices
   Suppose that CMS scans an industrial standard IT8 test target (in case of using digital camera, we need to control the surrounding light sources). Let CMS compare the distance between devices, and make a default profile for the future use.
3. The characterizations of color proofers or other output devices
   Pick and set a default profile from CMS for your color printer, printing condition, photo CD, or other output devices.
4. The printing outputs of test targets via selected output devices
   A presetting profile between monitors and printers (or scanners and printers) is used by CMS. A serious of device descriptive profiles can be used both in the image processing software and desktop publishing software. Some CMS even provides CMYK to preview color image on the monitors.

2.3 Gamut Mapping

Every device has its own color gamut. CIE color model is the largest gamut which could cover any device gamut, such as scanners, monitors, digital cameras, film recorders, CMYK color printers, proofers, etc. Usually the transformation between RGB and CMYK, CMS used to use the larger CIE gamut as the transformed space. For instance, we may transform the digital test target from an input device gamut to monitor gamut using default profile. Adobe Photoshop is able to be used to execute this instruction.

2.4 Color Appearance

Color appearances are not only affected by devices, but are also inferred by surrounding environments. There are CIECAM97 and CIECAM02 color appearance models proved by CIE to solve problems caused by surrounding environments.

3 ICC-based CMM

ICC Profile is a color characteristic description file which transforms color information between digital image peripheral devices, such as digital cameras, scanners, monitors, printers, etc. ICC Profile uses an international standard format whose size is very small, and is able to interoperate on different computer platforms and various operating systems. Through the optical measurement instruments and their associate software, by obtaining the standard color calibration cards, such as Kodak Q60R1, computers may compute and make the color transformation descriptive file. Then, it may be provided by some image processing software such as Adobe PhotoShop to correct and calibrate the right color while doing input and output functions. The detail is described as follows.

3.1 What is ICC?

Since the variety of input/output devices, though the information inside the image is not changed, different devices represent different color appearances. It is a critical issue during image color reproduction. How to make the scanned result consistent to the origin? How to print an image file with the same color using different printing equipments? How to print the same colors as seeing on the monitors? How to display the same colors in the different departments (especially important for editorial department) using different monitors?

These problems have been comprehensively solved by ICC industry standards. “International Color Consortium, ICC in short”, an organization was established by many well-known manufacturers, such as Adobe Systems Incorporation, Agfa-Gevaert N.V., Apple Computer, Eastman Kodak Company, Microsoft Corporation, Silicon Graphics Inc., Sun Microsystems Inc., etc in 1993. One of its missions is to integrate all digital image formats recently used, and to make a standard definition of device profile supporting device characterization for different platforms. Under this standard, input devices such as scanners, and digital cameras, display devices such as monitor, and output devices such as printer, and press, are calibrated by standard procedures, and then a color characterization descriptive file is built. The file is an ICC Profile that makes the color management possible using different color transformation models.

Before the investigation of ICC, the color matching procedures of CMS have to follow the proprietary rules made by individual manufacturers to build the characterization files of input/output devices. Although the direct transformation gives better color matching results, the growing of input/output devices raises the endless color calibration and color matching methods. Switching to XYZ color space may reach DIG (Device Independent Color) purpose. However, the lack of unified format makes it is not possible to apply on every individual device. ICC creates a characterization profile called Profile Connection Space (PCS), and asks all input/output devices follow the unified device characterization format, to reach universal color calibration and color matching, as shown in Figure 2.

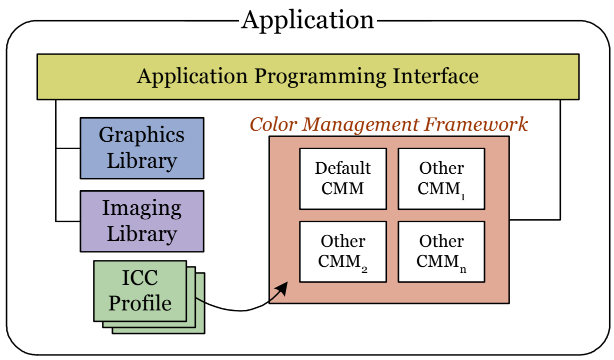


Figure 2. ICC-based CMM

3.2 The specification of ICC

The color characteristic of different devices is unified by ICC specification. Under the ICC specification, images are able to transfer from different brands, and to obtain a very good color reproduction appearance in different media. Since its superior performance, International Standards Organization (ISO) has put ICC for the color standard specification. You may use Color Sync (MAC) or ICM (MS WINDOWS) – some sorts of color management software to calibrate your color devices. The principle of ICC Profile is the color space transformation. In accordance with the numerical data of color characterization profile of each color device, we transform the color data from "Device-Dependent color space" to "Device-Independent color space". ICC uses "definition tag" to build the transformation model between each other by some mathematics model. Therefore, any color can be quantized, and any mathematics transformation equations can be implied. The stages of a color reproduction system are illustrated in Figure 3.

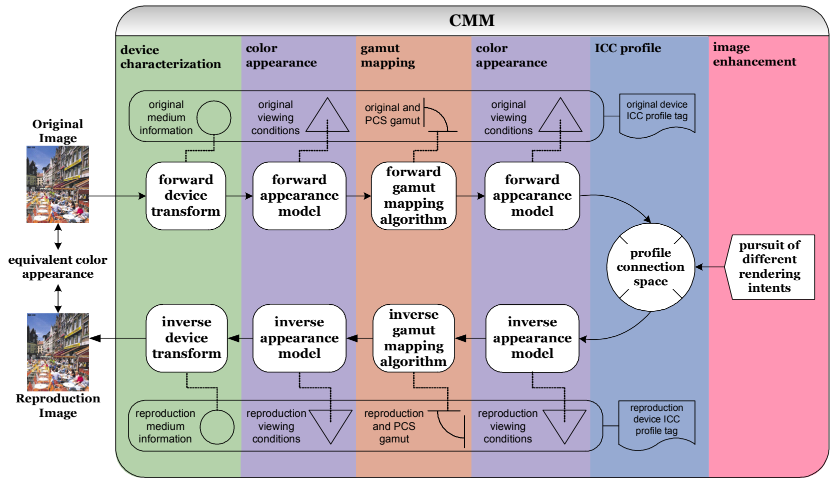


Figure 3. The Stages of a Color Reproduction System

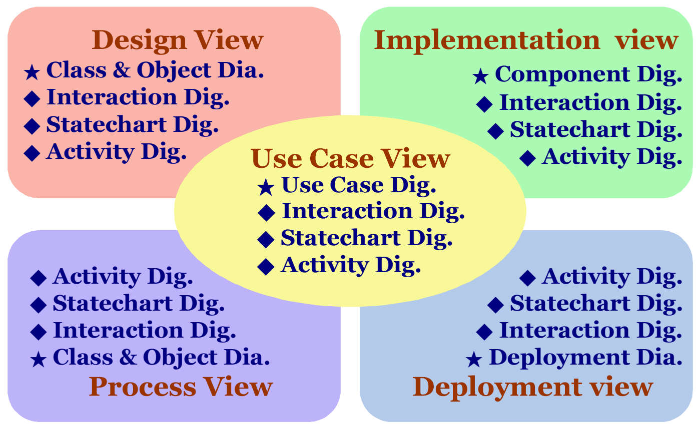
4 The Unified Modeling Language

Like to build a house or building, we may need all sorts of blueprints and models to communicate with one another. Because we might change the size, shape, and style of the house or building, even after we are starting to build the house or building.

In the fields of sociology, economics, and business management, we build models so that we can validate our theories or try to figure out new ones with minimal risk and cost. A model is a simplification of reality. We build models so that we can better understand the system that we are developing. Through modeling, we achieve four aims:

1. Models help us to visualize a system as it is or as we want it to be.
2. Models permit us to specify the structure or behavior of a system.
3. Models give us a template that guides us in constructing a system.
4. Models document the decisions we have made.

The Unified Modeling Language (UML) is a modeling language for specifying, visualizing, constructing, and documenting the artifacts of a system-intensive process. It was originally proposed by Grady Booch, James Rumbaugh, and Ivar Jacobson (Booch, 1998). The language has gained significant industry support from various organizations throughout the world and has already been adopted by the Object Management Group (OMG) as a standard. The architecture of the UML is illustrated in Figure 4.



★ represent the static aspect of its view

◆ represent the dynamic aspect of its view

"Dig." is the abbreviation of "Diagram"

Figure 4. The Architecture of the UML

In this research, a UML tool, Rational Rose (as shown in Figure 5), provided by Rational Corporation is used to design our framework. Using this tool, end users may easily see the functionalities of the system through use case view by use case diagrams. System analysts and designers may design the structure of the system through logical view by sequence diagrams, collaboration diagrams, and class diagrams. Programmers may manipulate the software, and system integrators may monitor the performance, scalability and throughput through component view by component diagrams. Finally, system engineers may realize the system topology, delivery, installation and communication through deployment view by deployment diagrams.

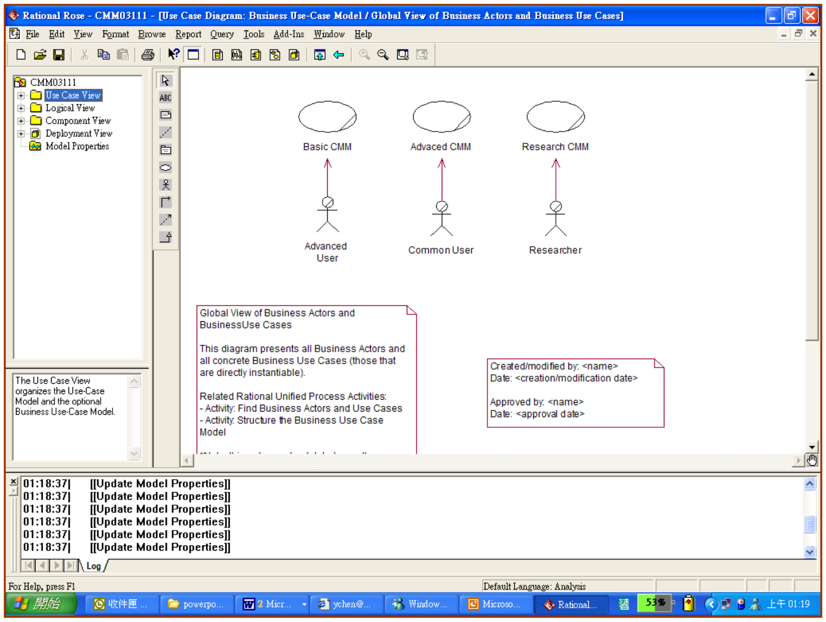


Figure 5. Rational Rose

5 Implementation

5.1 Scenarios of different Roles

This work followed our TAGA 2002 conference paper (Chen, 2002) presented last year. Three roles of Common Users, Advanced Users, and Researchers, were delivered. However, one more role called "Vendors" was added in this new framework.

For Common Users, one of the scenarios is to input an image for the testing device described in the stages of color reproduction system. In the Basic CMM architecture, users try to pick some existing profiles, i.e. simply to get the given data. For instance, device-dependent values (e.g. RGB or CMYK) of images are input and then transferred into device-independent PCS data, or vice verse. Therefore, Common Users only need to pick the existing ICC profiles to process images in question.

For Advanced Users, one of the scenarios is to train the functionalities of the system. For instance, they may input both device-dependent and colorimetric values from the ANSI IT8 testing target, and then build corresponding profiles using selected CMM algorithms including characterization, gamut mapping, and color appearance models. The other scenario is to generate some reports which shows whether the performance is good enough or how to obtain the better results while changing the selection a little bit. Moreover, we do provide a functionality to generate a new ICC profile for Advanced Users.

For Researchers, we try to give them more capability to operate the functionalities of system. Furthermore, they may borrow everything from the framework and build their own CMM upon the framework.

For Vendors, they may put their ICC profiles into the system directly. Therefore, the system may contain more ICC profiles for different devices from various manufacturers.

A summary of scenarios of different roles is described in Table 1.

Table 1. The scenarios of different roles

Roles	Scenarios
Common Users	input an image (RGB, PCS, etc) for the testing device
	select a default CMM or CMM combinations from existing ICC profiles for the testing device
	obtain an image (PCS, CMYK, etc) and generate a combined ICC-profile
Advanced Users	input device values and colorimetric values (IT8 test target) for the new device
	select CMM algorithms including characterization algorithm, gamut mapping algorithm, and color appearance model algorithm for the new device
	generate analytical reports for the selecting algorithms, including color difference values (average and maximum), and display image (sRGB)
	leverage some tuning tools to obtain better results (options)
	generate a new ICC-profile
Researchers	leverage framework to develop/add new functionality
	provide some tuning tools to obtain better results
Vendors	append or update ICC profiles

5.2 Use Case Diagrams

Use Case diagrams show us the actors, the use cases, and their relationship in our system. An actor is someone or something that interacts with the system being built. A use case is a high-level description of functionality that the system will provide. In other words, a use case illustrates how someone might use the system. Actors are something outside the designated system, and use cases are something inside the designated system

According to the CMS scenarios described in the previous sub-section. We design four actors, including Common User, Advanced Users, Researchers, and Vendors in the Use Case diagram. Common Users play the roles of color setting analyzers in printing factories, professional graphics designers, or the beginners who want to study CMS. Furthermore, Advanced Users play the roles of the experts who are understood well the CMS. Researchers are just like the professors or researchers who investigate the CMS in the institutes. Finally, Vendors play the roles of equipment supplier.

Moreover, six use cases including Vendor Mode Setting, Basic Mode Setting, Image Generator, Data Transformation, and ICC Profile Generator are designed to be used by different actors. The content of each use case is described in the following paragraphs:

Vendor Mode Setting: The purpose of this use case is to provide the ICC profiles for new devices or to update the ICC profiles which have been stored inside the system. This use case is used by Vendors actor, and the information flows are in both directions— from the Vendors to the Vendor Mode Setting and back again — where the arrow indicates who initiates the communication.

Basic Mode Setting: It is used by Common Users and includes three other use cases to provide more functionalities— Image Generator, Data Transformation, and ICC profile Generator described later on. The primary work is to correct the color of wanted CMS image. Actually it provides a user interface letting actors input an image and an ICC-based CMM. The image is made by CMYK, RGB, PCS, etc from the testing device and the default CMM or a CMM combination from existing ICC profiles for the testing device selected by Common Users. Then, it transforms the image color data from device values to colorimetric values and generates an ICC profile to be used later or a sRGB image to be displayed on the monitors for evaluating the results of processing.

Image Generator: This use case is like a program library but always provides a function to generate images. There are two dashed arrow relationships with the stereotype "include" both from Basic Mode Setting and Advanced Mode Setting. The functionality of it is to generate, display, or save a standard sRGB image.

Data Transformation: It is also a functional use case, which provides the most important work in CMM. The kernels of this use case mentioned early are device characterization, color appearance model, and gamut mapping. The method of processes depends on ICC profile where user selected, but the sequence of processes depends on CMM itself.

ICC Profile Generator: The CMM system can output an ICC profile for later use. The file format follows the ICC specification.

Advanced Mode Setting: Like Common Mode Setting, this use case almost does the same work, but it more focuses on analyzing the result of a default CMM or CMM combination whether it is good enough after processing.

The User Case Diagram of CMS is illustrated in Figure 6.

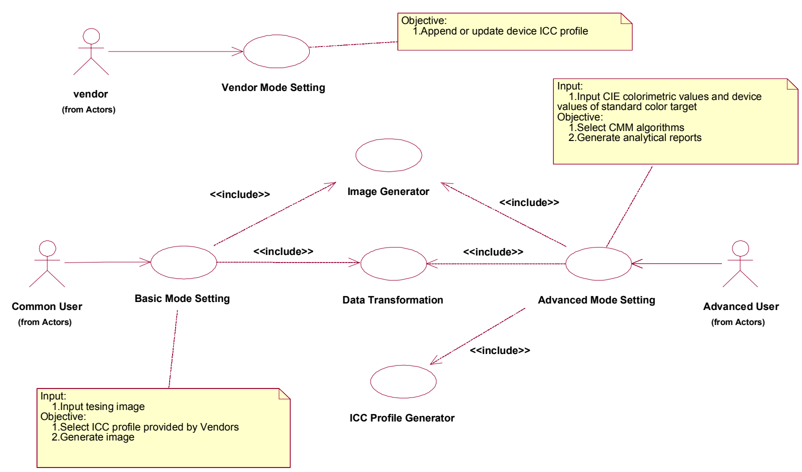


Figure 6. The use case diagram of CMS

5.3 Interaction Diagrams

An Interaction diagram shows one of the flows inside a use case step-by-step. It describes which objects are needed in the flow, which messages are sent by objects, which flows are initiated by actors, and which sequences are followed by the messages. An object is something that encapsulates information and behavior. A message is a communication between objects in which one object (the Client) asks another object (the Supplier) to do something. In the time of generating codes, all of objects should be mapped to classes and messages should be mapped to an operation of the "Supplier" class, which would be translated to function calls.

In the CMS, for instance, there are several alternative flows inside the Basic Mode Setting use case. Therefore, there are several Interaction diagrams for this use case. There is a Basic Mode Interaction diagram, which shows what happens when input a RGB image. Furthermore, there are additional diagrams showing the alternative flows, such as what happens when a Common User inputs a PCS image, or what happens when an input image has an incorrect file format, and so on. All of the different scenarios that CMS needs to implement are drawn in the Interaction diagram.

Two types of Interaction diagrams, Sequence diagram and Collaboration diagram, are used in the UML. A Sequence diagram is organized by time and read from the top to the bottom. A Collaboration diagram shows the same information, but is organized differently. It focuses more on the relationships between the objects. There are three Interaction diagrams to be implemented in the Basic Mode Setting use case, i.e., Basic Mode, Basic Data Transformation, and Basic Image Generator.

Basic Mode: By looking at the objects and messages from the top to the bottom of the Basic Mode sequence diagram, shown in the figure 7. The objects participate in the flows are shown in rectangles at the top of the diagram. The object, Form, is a client page that is displayed to the end user— Common Users. The remaining objects, Image and SHUICC, constitute the server-side pages, and the Control object is responsible for coordinating the efforts of other classes.

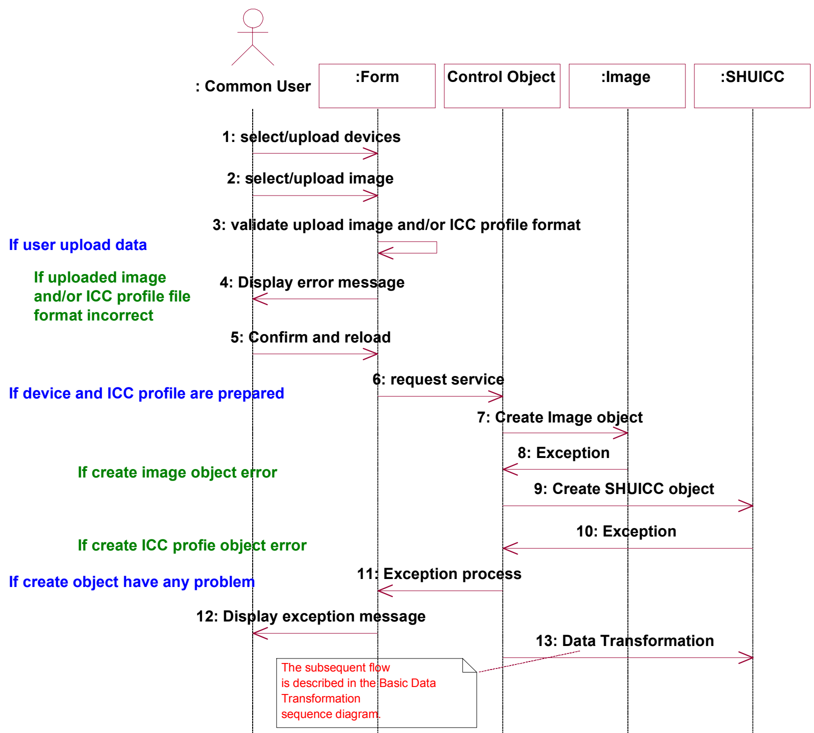


Figure 7. Sequence Diagrams - Basic Mode

First of all, the Common User selects a device and input an image file supplied by the Form object, and then the input image file format is validated by the Image object. If incorrect, the Image object will send a message to the Form object called Exception by the program, and then display an error message to the Common User. Otherwise, image file format is correct, and the Image object will pick an ICC profile according to the selected device by Common User and then the CMS does the pixel data transformation according to the other Data Transformation sequence diagram.

If we want to see the relationships between objects in the Basic Mode, we may see the other Interaction diagram, the Basic Mode collaboration diagram shown in the figure 8.

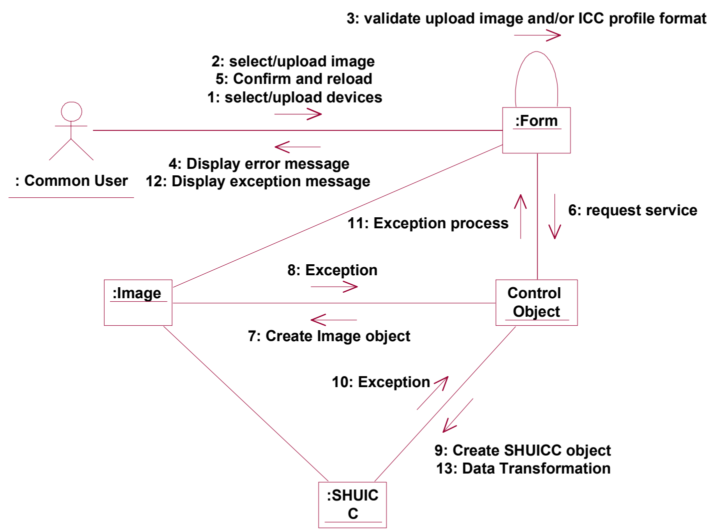


Figure 8. Collaboration Diagrams – Basic Mode

Basic Data Transformation: In this sequence diagram, there are two processes. One is forward processing, and the other is inverse processing. The forward processing is to transform the device color values to the standard colorimetric values, and inverse processing does the same thing but from the colorimetric values to the device color values. The processes follow the color reproduction system mentioned early, and the detail is shown in the figure 9.

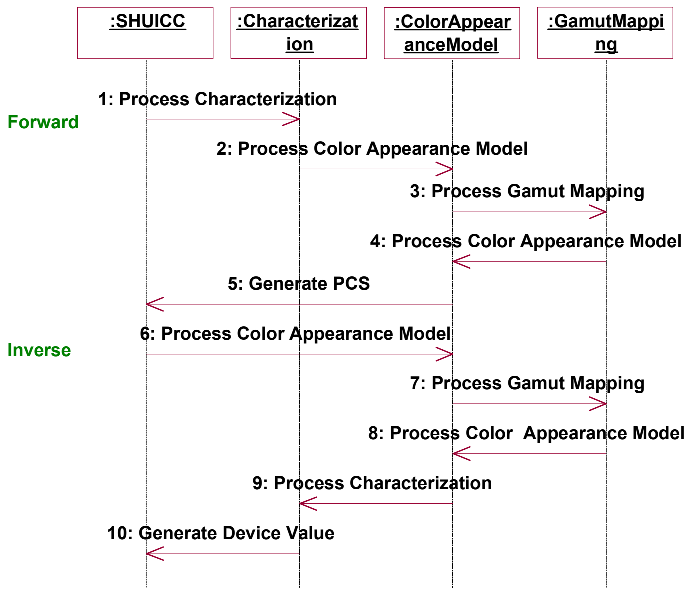


Figure 9. Sequence Diagrams – Basic Data Transformation

Basic Image Generator: When the pixel data transformation is finish, the next process is to show the result in sRGB monitor and may need to save the processed image as shown in figure 10.

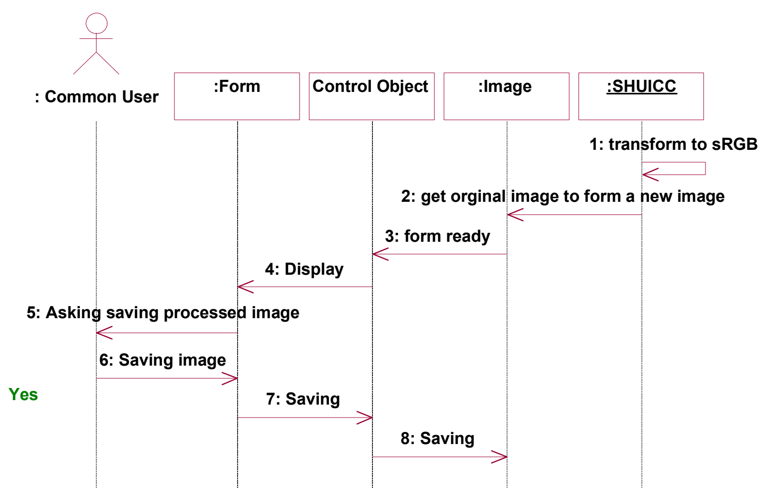


Figure 10. Sequence Diagrams - Basic Image Generator

5.4 Class Diagrams

In the previous section, it can be seen that the objects interact with each others in order to describe their functionalities. In this section, we take a close look at the objects themselves. A Class diagram is used to display some of the classes and package in the system. It gives a static picture of the pieces in the system and of the relationships between them. A class is something that encapsulates information and behavior of objects.

One guideline in object-oriented programming is to separate the implementation of a class from its interface. Most object-oriented languages now support the concept of an interface, which contains only the method signatures (without the implementation) for a class.

Classes are abstracted based on the analysis of the system. While use cases specify the operations of the users on the system, objects depict the entities that contain such operations. Apparently the operations concerning device characterization belong to one class, yet image transformation belongs to another. The following class diagram (figure 11) shows the fundamental classes and their relationships.

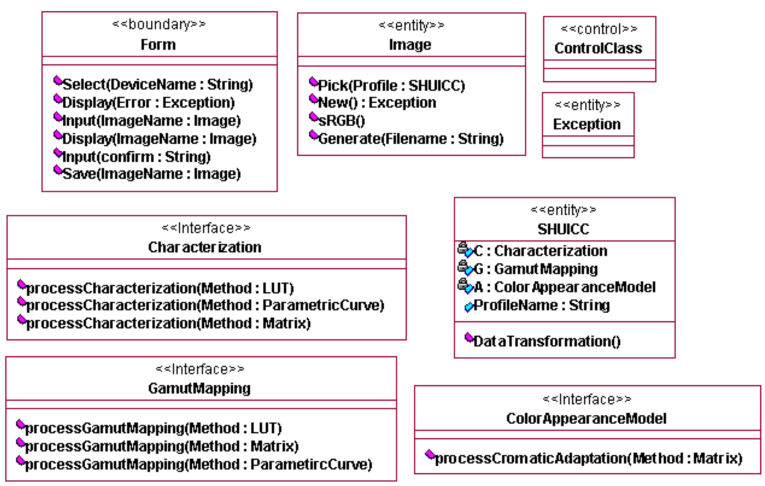


Figure 11. The class diagram of the ICC-based CMS

Several interfaces specify the externally visible operations of a class and/or component, and have no implementation of its own are listed as follows.

- Characterization
  ProcessCharacterization (Method:LUT)
  ProcessCharacterization (Method:Matrix)

ProcessCharacterization (Method:ParametricCurve)

- Gamut Mapping
  ProcessGamutMapping (Method:LUT)
  ProcessGamutMapping (Method:Matrix)
  ProcessGamutMapping (Method:ParametricCurve)
- Color Appearance Model
  ProcessCromaticAdaptation (Method:Matrix)

Researchers may borrow this framework and implement their own CMS following these interfaces. Later on, everyone may share each others' modules because of the compatibility. However, the scope of how to construct a compatible CMS is not the issue covered in this research.

6 Conclusions and Future Work

A color management framework is designed by the UML, and work is done listed as follows.

- Use Case Diagrams in Use Case View
- Sequence Diagrams, Collaboration Diagrams and Class Diagrams in Logical View
- Interfaces of Color Management Modules
- A programming language independent schema

For the future work, the scenario of Researchers such as providing a tuning tool has not done in this research. It is worth of study later on. Another important issue is to provide an XML-compliant color management system. It is a standard letting Researchers simply to build their own CMM on their side, and then to share the information, something like distributed objects of color management modules, to each other just by the Internet. Two tasks are going to study after this research.

First of all is to fulfill the functionality of the role of researchers:

- to leverage framework to develop/add new functionality
- to provide some tuning tools to obtain better results

Next is to implement an XML-compliant Color Management System:

- to design Distributed Objects of Color Management Modules
- to leverage Universal Description, Discovery and Integration (UDDI) of Web-services

References

ANSI IT8.7/1

1993 Graphic Technology- Color Transmission Target for Input Scanner Calibration, American National Standards Institute, Inc.

Y. L. Chen, M. C. Lo, T. Q. Wen and C. C. Yu

2002 *Cross-Media Color Management via the Internet*, TAGA 2002 54th Annual Technical Conference, Asheville, NC, U.S.A., April 14-17.

E. J. Giorgianni and T. E. Madden

1998 *Digital Color Management: Encoding Solutions*, Addison-Wesley.

G. Booch, J. Rumbaugh, and I. Jacobson

1998 *The Unified Modeling Language User Guide*, Addison-Wesley.

ICC

2001 *Specification: 2001-04*, File Format for Color Profiles, http://www.color.org.

M. R. Luo, G. Gui, and B. Rigg

2001 Corresponding-Colour Datasets, *Color Research and Application*. **26**:340-350.

N. Moroney, M. D. Fairchild, R. W. G. Hunt, Changjun Li, M. R. Luo, and Todd Newman

2002 The CIECAM02 Color Appearance Model, *Proc. 10th IS&T/SID Color Imaging Conference*, pp.23-27.